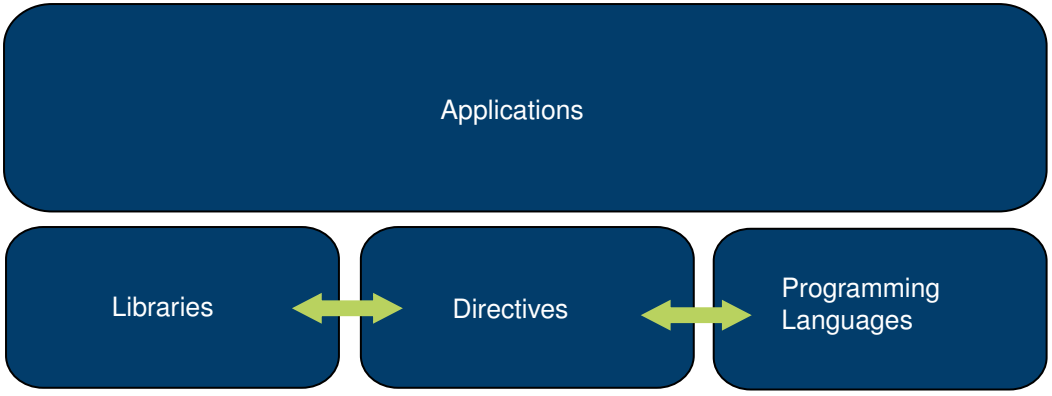
JUWELS BOOSTER PORTING WORKSHOP
GPU-Accelerated Libraries

January 22, 2021 | Kaveh Haghighi Mood | JSC

OVERVIEW

- Introduction
- Communication libraries
- Core libraries
- Math libraries
- Domain Specific Libraries

ACCELERATION POSSIBILITIES



WHY LIBRARIES?

Programming GPUs is easy: No need to reinvent the wheel, use libraries!

WHY LIBRARIES?

Programming GPUs is easy: No need to reinvent the wheel, use libraries!

WHY LIBRARIES?

Programming GPUs is easy: No need to reinvent the wheel, use libraries!

- Delivers good performance

WHY LIBRARIES?

Programming GPUs is easy: No need to reinvent the wheel, use libraries!

- Delivers good performance
- Portability

WHY LIBRARIES?

Programming GPUs is easy: No need to reinvent the wheel, use libraries!

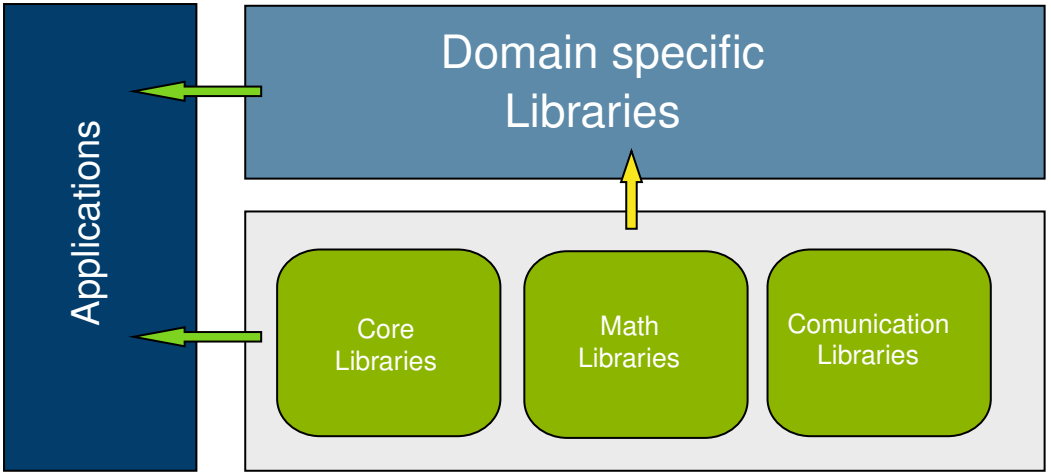
- Delivers good performance
- Portability
- Optimizations for different architectures $\longrightarrow$ Not your responsibility

WHY LIBRARIES?

Programming GPUs is easy: No need to reinvent the wheel, use libraries!

- Delivers good performance
- Portability
- Optimizations for different architectures $\longrightarrow$ Not your responsibility
- Testing and bug fixing $\longrightarrow$ Not your responsibility

GPU LIBRARIES



COMMUNICATION LIBRARIES

- CUDA aware MPI: No need to stage GPU buffers through host memory
- NVSHMEM: Combines the memory of multiple GPUs into a partitioned global address space
- NCCL: Collective communication routines for GPUs

CORE LIBRARIES

- CUB: C++ Template library primitives
  - Warp-, block-, and device-wide parallel primitives
  - Parallel sort, prefix scan, reduction, histogram
  - Dynamic parallelism

CORE LIBRARIES

- CUB: C++ Template library primitives
  - Warp-, block-, and device-wide parallel primitives
  - Parallel sort, prefix scan, reduction, histogram
  - Dynamic parallelism
- Thrust: Parallel C++ template algorithms similar to the C++ standard library for the GPUs
  - Based on CUB
  - Higher level library
  - Usually called from host
  - Containers (host and device vectors)
  - Algorithms (sort, search, reductions, random numbers, ...)

MATH LIBRARIES

MATH LIBRARIES

cuRAND

CUDA pseudorandom generator:

- Host and device random number generator
- Nine type of generators include Mersenne Twister, Xorshift and MRG
- Supports Uniform, Poisson and Normal distributions

How to use:

1. Create a new generator:

```
curandCreateGenerator(&g, CURAND_RNG_PSEUDO_DEFAULT);
```

2. Set the generator options (seed, offset, ...):

```
curandSetPseudoRandomGeneratorSeed(g,1648195);
```

3. Generate random numbers:

```
curandGenerateUniformDouble(g, x_d, sampleSize);
```

4. Clean up:

```
curandDestroyGenerator(g);
```

TASK1

Monte Carlo integration with cuRAND

In this exercise, we'll use cuRAND to generate random numbers for Monte Carlo integration:

$$I = \int_a^b f(x)dx = \lim_{N\to\infty} \frac{b-a}{N} \sum_{i=1}^{N} f(x_i). \tag{1}$$

Crude MC integration algorithm:

1. Draw the sample out of the uniform distribution
2. Calculate function values with the sample
3. Estimate the results $I_{es} = \frac{b-a}{N} \sum_{i=1}^{N} f(x_i)$

TASK1

Monte Carlo integration with cuRAND

- Navigate to:

  `GPU-Optimised-Libraries/exercises/tasks/cuRAND`

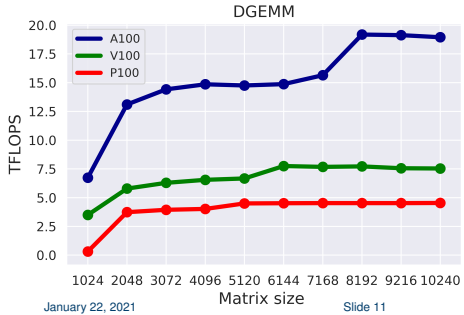
- Look at `Instructions.ipynb` for instructions
- See the cuRAND documentation for further information: URL
- For the Fortran interface check URL
- Call `source setup.sh` to load the modules of this task into your environment

MATH LIBRARIES

cuBLAS

GPU-accelerated implementation of the basic linear algebra subroutines:

- Complete 152 BLAS routines
- Multi-GPU support
- Supports CUDA streams
- Uses Tensor cores if available
- cuSPARSE provides similar functionality for sparse matrices



JÜLICH Forschungszentrum

MATH LIBRARIES

cuBLAS

How to use:

1. Create a handle:

```
cublasHandle_t handle;
cublasCreate(&handle);
```

2. Copy the data to device:

```
cublasSetVector(n, sizeof(x[0]), x, 1, d_x, 1);
```

3. Call cuBLAS:

```
cublasSaxpy(n, a, d_x, 1, d_y, 1);
```

4. Copy the data back to host:

```
cublasGetVector(n, sizeof(y[0]), d_y, 1, y, 1);
```

5. Clean up:

```
cublasDestroy(handle);
```

TASK2

cuBLAS

Matrix-matrix multiplication:

- Location of code: GPU-Optimised-Libraries/exercises/tasks/cuRAND
- Check cuBLAS documentation for details on cublasDgemm()
- For the Fortran interface check URL
- Look at Instructions.ipynb Notebook for instructions
- implement call to double-precision GEMM of cuBLAS

MATH LIBRARIES

cuFFT

GPU-accelerated implementation of FFT:

- 1D, 2D, 3D transforms
- Half, single and double precision, complex and real data types support
- Supports CUDA streams
- Batch execution
- FFTW like API
- Multi GPU support
- Device Extensions (cuFFTDx)

MATH LIBRARIES

cuFFT

How to use:

1. Create a plan:

```
cufftHandle plan;
cufftCreatePlan(plan, Nx, Ny, CUFFT_Z2Z);
```

2. Copy the data to device
3. Call cuFFT:

```
cufftExecZ2Z(plan, src, tgt, CUFFT_FORWARD);
```

4. Copy the data back to host
5. Clean up:

```
cufftDestroy(plan);
```

TASK3

cuFFT

Use cuFFT to find the components of a signal:

- Location of code: GPU-Optimised-Libraries/exercises/tasks/cuFFT
- Check cuFFT documentation for details on `cufftPlan1d` and `cufftExecD2Z`
- For the Fortran interface check URL
- Look at Instructions.ipynb notebook for instructions
- implement call to cuFFT to perform a real to complex Fourier transformation
- use Instructions.ipynb notebook to visualize results

MATH LIBRARIES

cuSOLVER

Collection of dense and sparse direct linear solvers and Eigen solvers:

- Similar to LAPACK
- Both dence and sparse solvers
- Multi GPU support
- Tensor core support
- Mixed-precision iterative refinement using tensor cores

MATH LIBRARIES

cuTENSOR

Tensor Linear Algebra on GPUs:

- Direct Tensor Contraction
- Reduction
- Elementwise Operations
- Mixed precision support
- Multi GPU support
- High level interfaces (cutensorex) for Fortran intrinsic array functions

MATH LIBRARIES

- Magma
  - LAPACK for heterogeneous architectures
  - Multiple precision arithmetic support
  - Using both multicore CPUs and GPUs
- ArrayFire
  - Vector Algorithms
  - Linear Algebra
  - FFT
- SLATE
  - ScaLAPACK replacement
  - BLAS
  - Linear solvers

MATH LIBRARIES

- heFFTe $\longrightarrow$ FFTs for Exascale
- DBCSR $\longrightarrow$ Sparse Matrix Multiplication Library
- AmgX $\longrightarrow$ Algebraic Multigrid solver
- COSMA $\longrightarrow$ GPU-accelerated, matrix-matrix multiplication
- deal.II $\longrightarrow$ PDE solver
- PETSc $\longrightarrow$ Portable, Extensible Toolkit for Scientific Computation Solvers for
  - Nonlinear time-dependent differential equations
  - Algebraic equations
  - Numerical optimization

DOMAIN SPECIFIC LIBRARIES

Search before implementation. Use the work of other people in your discipline

- Physics libraries
  - Sirius $\longrightarrow$ electronic structure calculations
  - QUDA $\longrightarrow$ lattice QCD
  - OpenMM $\longrightarrow$ molecular simulation

DOMAIN SPECIFIC LIBRARIES

Search before implementation. Use the work of other people in your discipline

- Physics libraries
  - Sirius $\longrightarrow$ electronic structure calculations
  - QUDA $\longrightarrow$ lattice QCD
  - OpenMM $\longrightarrow$ molecular simulation
- Climate and weather
  - GridTools $\longrightarrow$ performance portable lib for stencil-like patterns
  - GHEX $\longrightarrow$ halo-exchange operations on variety of grids/meshes
  - GCL $\longrightarrow$ The Generic Communication Library for halo exchange pattern of communication

DOMAIN SPECIFIC LIBRARIES

Search before implementation. Use the work of other people in your discipline

- Physics libraries
  - Sirius $\longrightarrow$ electronic structure calculations
  - QUDA $\longrightarrow$ lattice QCD
  - OpenMM $\longrightarrow$ molecular simulation
- Climate and weather
  - GridTools $\longrightarrow$ performance portable lib for stencil-like patterns
  - GHEX $\longrightarrow$ halo-exchange operations on variety of grids/meshes
  - GCL $\longrightarrow$ The Generic Communication Library for halo exchange pattern of communication
- Computational Neuroscience
  - Arbor
  - GeNN

- Data science, Deep/Machine Learning Libraries
  - Rapids
  - cuDNN
  - TensorFlow
  - PyTorch
  - Chainer ...
- Image, video and audio processing and analysis Libraries
  - nvJPEG, NVIDIA Video Codec SDK, NVIDIA Optical Flow, NVIDIA Video Codec SDKs
  - FFmpeg

    ...

LIBRARIES

Libraries for AMD GPUs

CUDA	HIP
cuBLAS	rocBLAS
cuRAND	rocRAND
cuFFT	rocFFT
Thrust	hipThrust
CUB	rocPRIM
cuSolver	rocSolver
AmgX	rocALUTION
cuSPARSE	rocSPARSE
cuDNN	MIOpen
NCCL	RCCL

- ROCm Libraries: URL

RESOURCES

- CUDA toolkit documentation
- Fortran Interfaces
- Examples:
  `NVHPC-INSTALLDIR/arch/version/examples/CUDA-Libraries`
  `CUDA-INSTALLDIR/samples/7_CUDALibraries`

RESOURCES

- CUDA toolkit documentation
- Fortran Interfaces
- Examples:
  `NVHPC-INSTALLDIR/arch/version/examples/CUDA-Libraries`
  `CUDA-INSTALLDIR/samples/7_CUDALibraries`

Thank you for your attention!